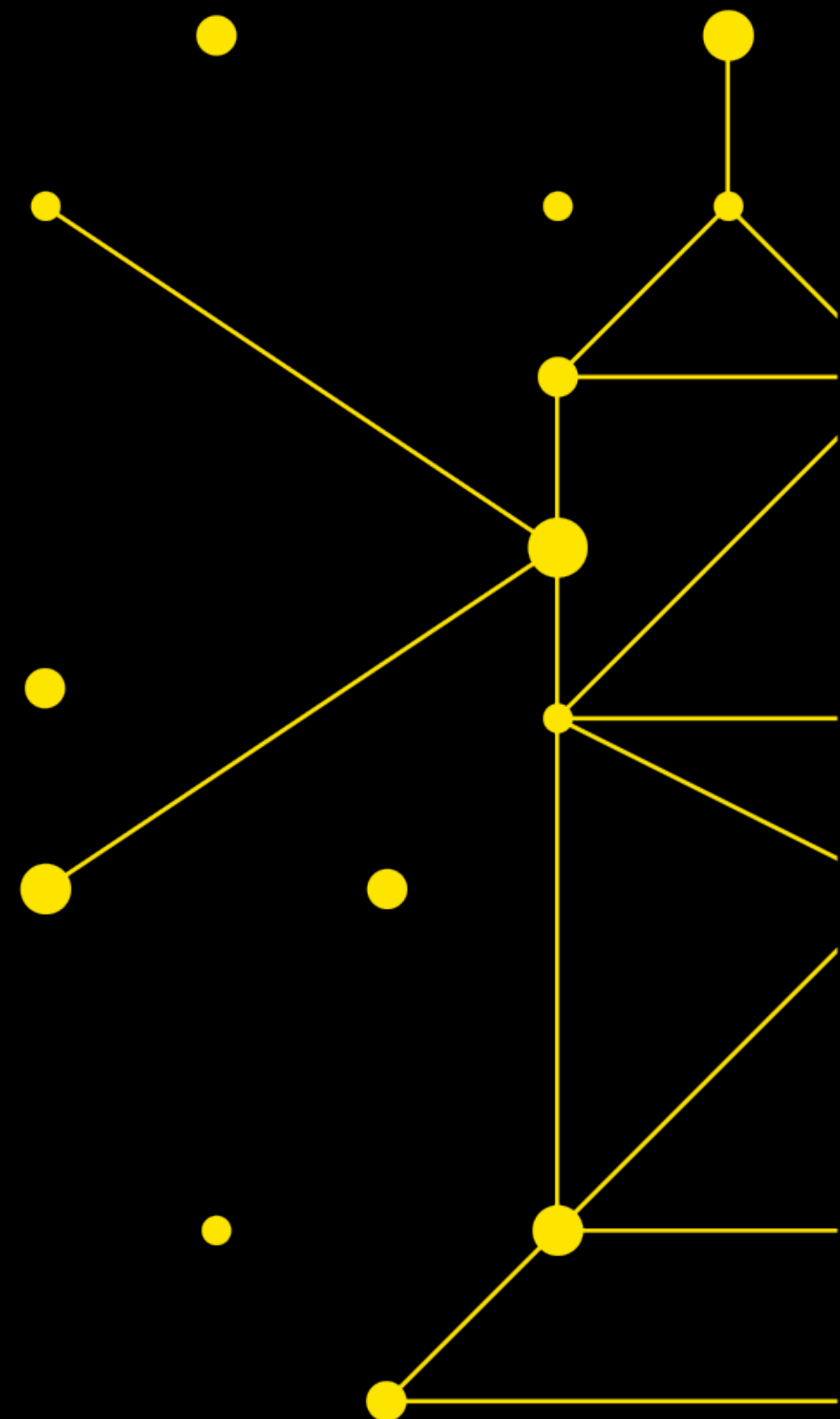
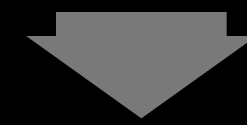
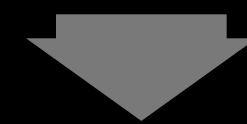
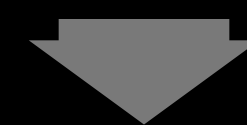
쿠키런 로그 시스템

바쁘고 가난한 개발자를 위해

데브시스터즈
임중근

1. 발표 소개

개요

게임 로그 시스템의 필요성

쿠키런 로그 시스템의 철학

쿠키런 로그 시스템의 구조

활용사례

발표자 소개

임중근

데브시스터즈 2014~현재

게임 서버
개발 및 운영

NCSOFT 2007~2012

Hadoop 기반 MMORPG
로그 시스템 구축 및 운영

분산 스트리밍 SQL 쿼리
엔진 개발 및 운영

2. 쿠키런 서비스 개요

2. 쿠키런 서비스 개요

쿠키런

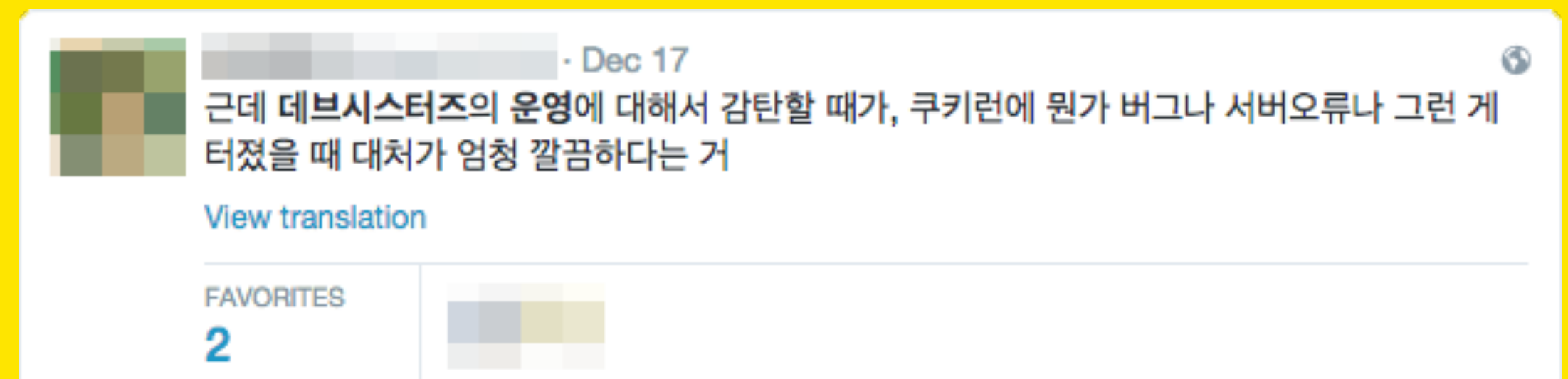
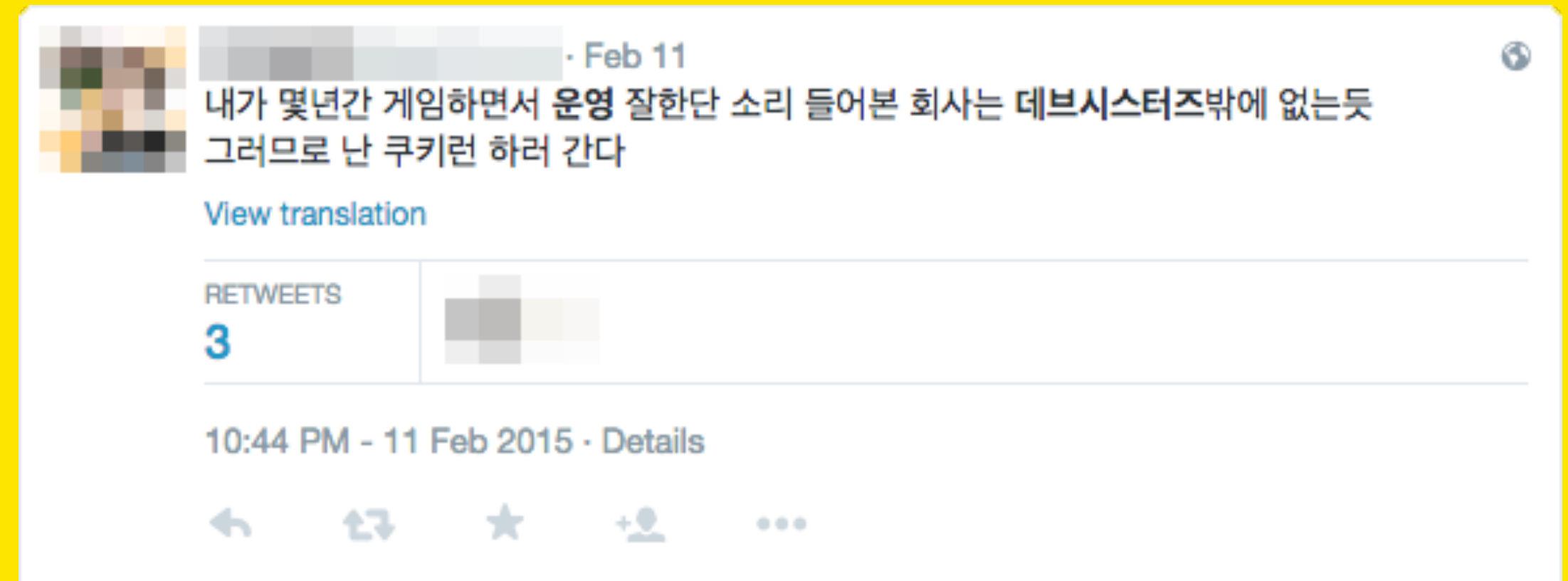
카카오 / 라인을 통한 **글로벌** 서비스

DAU **1천만** 기록

7천만 누적 다운로드 기록

2주년 이벤트를 치른 **장수 모바일 게임**

2. 쿠키런 서비스 개요 쿠키런 운영

- 운영 잘 하는 게임
- 장애에 대한 **확실한 보상**
- **친절하고 명확한** 고객 응대
- 유저와의 **신뢰** 구축

2. 쿠키런 서비스 개요

쿠키런 로그 시스템

- 쿠키런 운영의 경쟁력
- **총 100TB 이상**의 로그 삭제 없이 전부 보관
- 한 유저의 전체 플레이 기록을 **3초** 안에 열람
- Hadoop 클러스터로 **빠른 대규모 분석** 가능
- **초기 투자비용 제로, 월 수십만원** 대의 유지 비용

3. 과거 로그를 왜 저장하죠?

3. 과거 로그를 왜 저장하죠? 법무 대응

아이템이 없어졌다. 물어내라.

정확한 전후 정황이 패소를 막는다.

3. 과거 로그를 왜 저장하죠? 고객 문의 대응

아이템이 없어졌어요, 보상해 주세요.

묻지도 따지지도 않고 보상해주면?

>>> 고객 대응이 마비된다.

3. 과거 로그를 왜 저장하죠? 통계 및 지표

DAU, MAU, WAU, PU, …

새로이 개발한 지표를 과거에 대해서 뽑는다.

지표 개발이 쉽고 빨라진다.

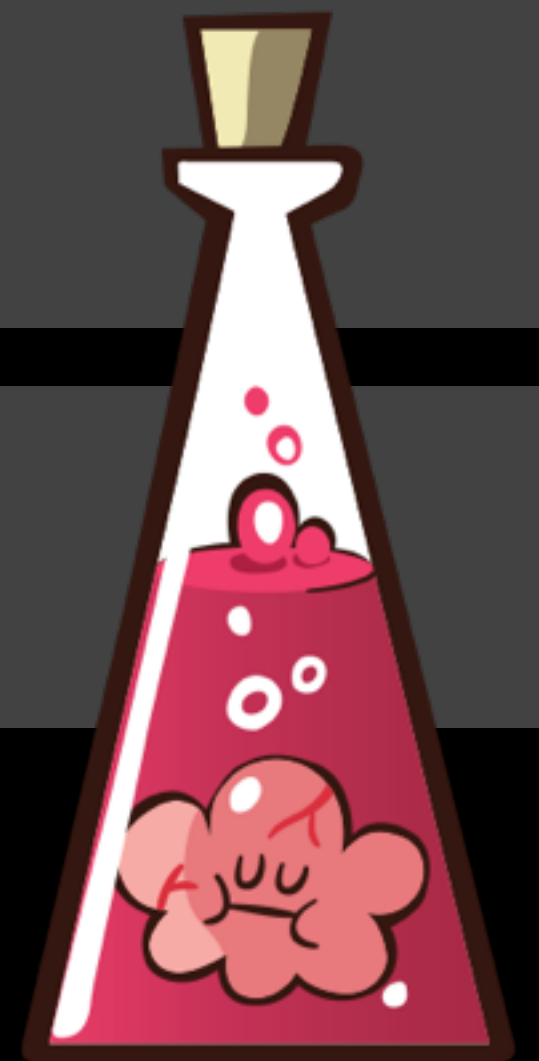
4. 쿠키런 로그 시스템의 철학

쿠키런 게임 서버

Java, Spring MVC, Tomcat

Couchbase, MySQL, Redis

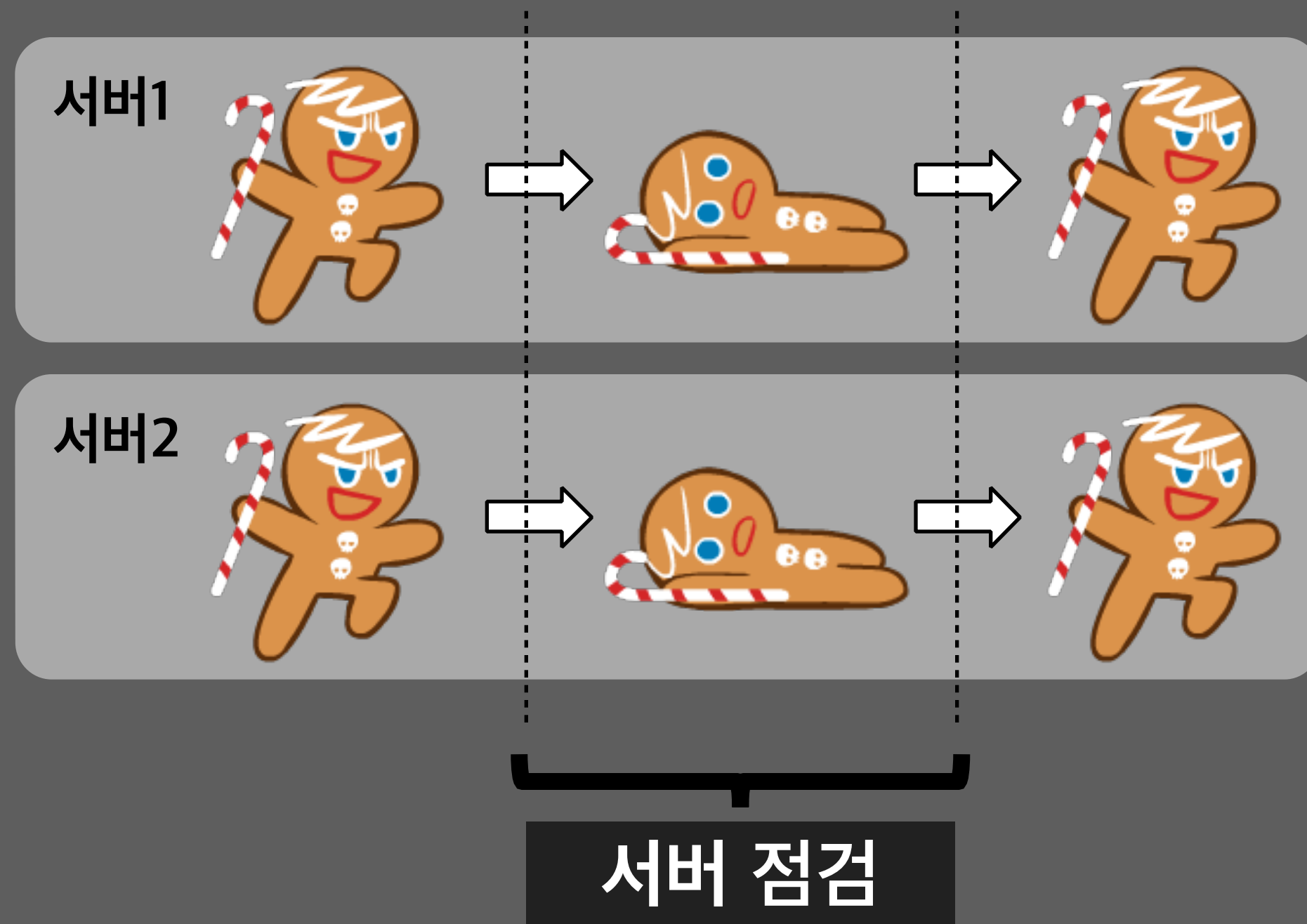
아마존 웹 서비스

24/7 무점검 무중단 서비스를 지향

4. 쿠키런 로그 시스템의 철학

무점검 서버 패치

일반적인 경우

쿠키런의 경우

서버 컴포넌트의 조건

- 모든 서버 프로세스는 원치 않을 때 죽는다
  – 예상치 못한 시점에 죽기도 하지만
  – 긴급 보안 패치가 있기도 한다.
- 언제나 메인터넌스와 패치를 가능하게
  – 왜냐하면…

4. 쿠키런 로그 시스템의 철학 ‘바쁘다’

- 컨텐츠 개발하느라 바쁘다.
- 발생한 문제는 그 때 그때 해결하고 잊자.
- 주기적 메인터넌스 최소화
- 개발자 컴퓨터에서 쉽게 접근 가능하게

4. 쿠키런 로그 시스템의 철학 '가난하다'

- 게임 서버 비용보다 로그 서버 비용이 비싸면 곤란
- 무리한 초기 투자를 할 수 없다.
  – 게임의 성패는 아무도 모른다.
- 생각보다 훨씬 저렴한 유지비로 구축할 수 있다

5. 쿠키런 로그 시스템의 개요

쿠키런 로그

- 매일 200~400GB 의 JSON 텍스트 형식
  - 2년치 총량 100TB 이상
- 비정형 스키마
- 왜 바이너리 포맷 안 쓰나요? → 바빠요
  - 로그를 추가할 때마다 전체 시스템을 업데이트하기 싫어요

쿠키런 로그 시스템 구성

Tomcat

Kafka

Real Time
Log Server
LevelDB

Amazon S3
(Daily Log)

CS Server

Log Broker

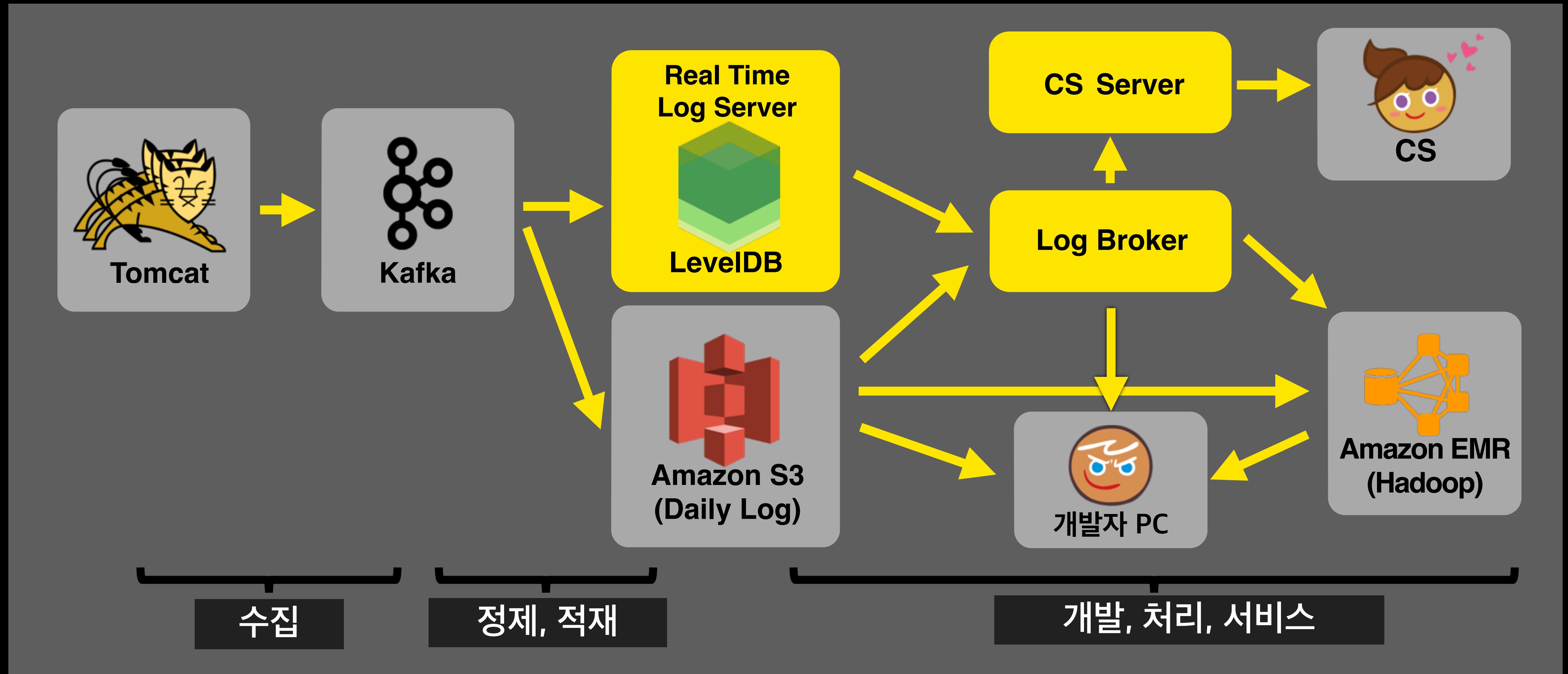
CS

개발자 PC

Amazon EMR
(Hadoop)

수집

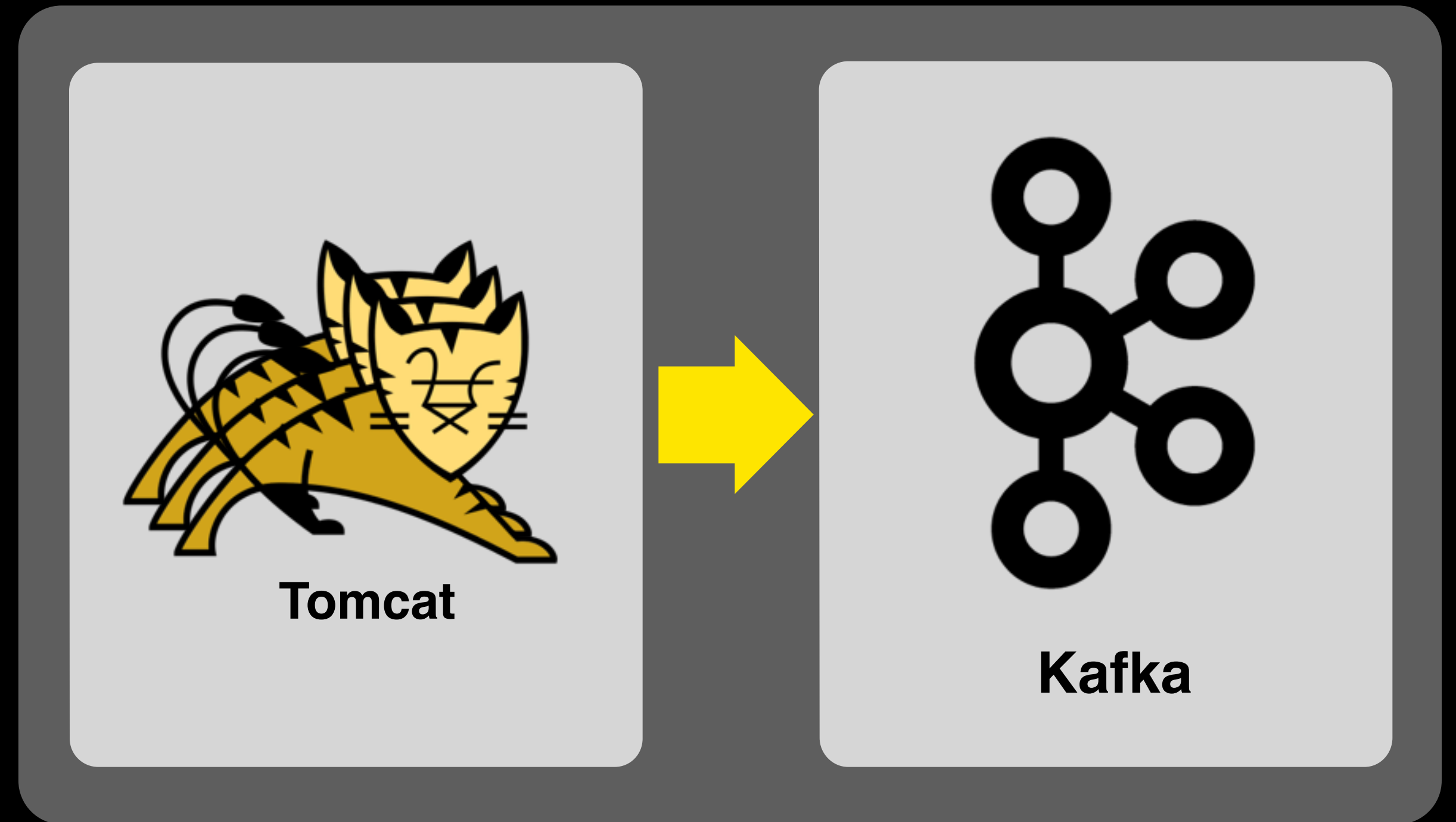
정제, 적재

개발, 처리, 서비스

6. 로그 수집 - Apache Kafka

6. 로그 수집 Apache Kafka

로그 수집

- Apache Kafka 사용
- 분산 메세징 시스템
  - Publish / Subscribe
- 일별 토픽, 나흘치 저장

Kafka의 장점

- 메세지를 복제하여 저장
  - 노드 장애에 강함
  - 서버 업데이트 및 재부팅이 상시 가능
- 단순한 저장소 구조, 고장이 잘 안 난다
- 높은 읽기 / 쓰기 대역폭

Logstash → Kafka

- 컴포넌트를 멈추면 어딘가 문제가 일어난다.
  - 프로세스가 죽을 때 로그 유실 발생
- Redis를 내렸다 올리면 큰일나요
  - Redis 에 긴급 점검 스케쥴이 걸리면 워크샵 때 못 놀아요
- Logstash는 Push 방식, Kafka는 Pull 방식
  - 로그를 읽는 프로세스를 내렸다 올릴 수 있다.

Kafka 사용의 주의점

- 메세지 중복은 피할 수 없다.
  - Two Generals' Problem
  - 기록 성공 응답을 못 받으면 중복 발생
- 중복 제거는 정확한 통계를 위해 필수

7. 로그 백업 - Amazon S3

로그 백업

- 자기 테입 백업?
- NAS?
- MySQL + 일별 백업?
- Hadoop 클러스터?
- HBase / Cassandra 클러스터?

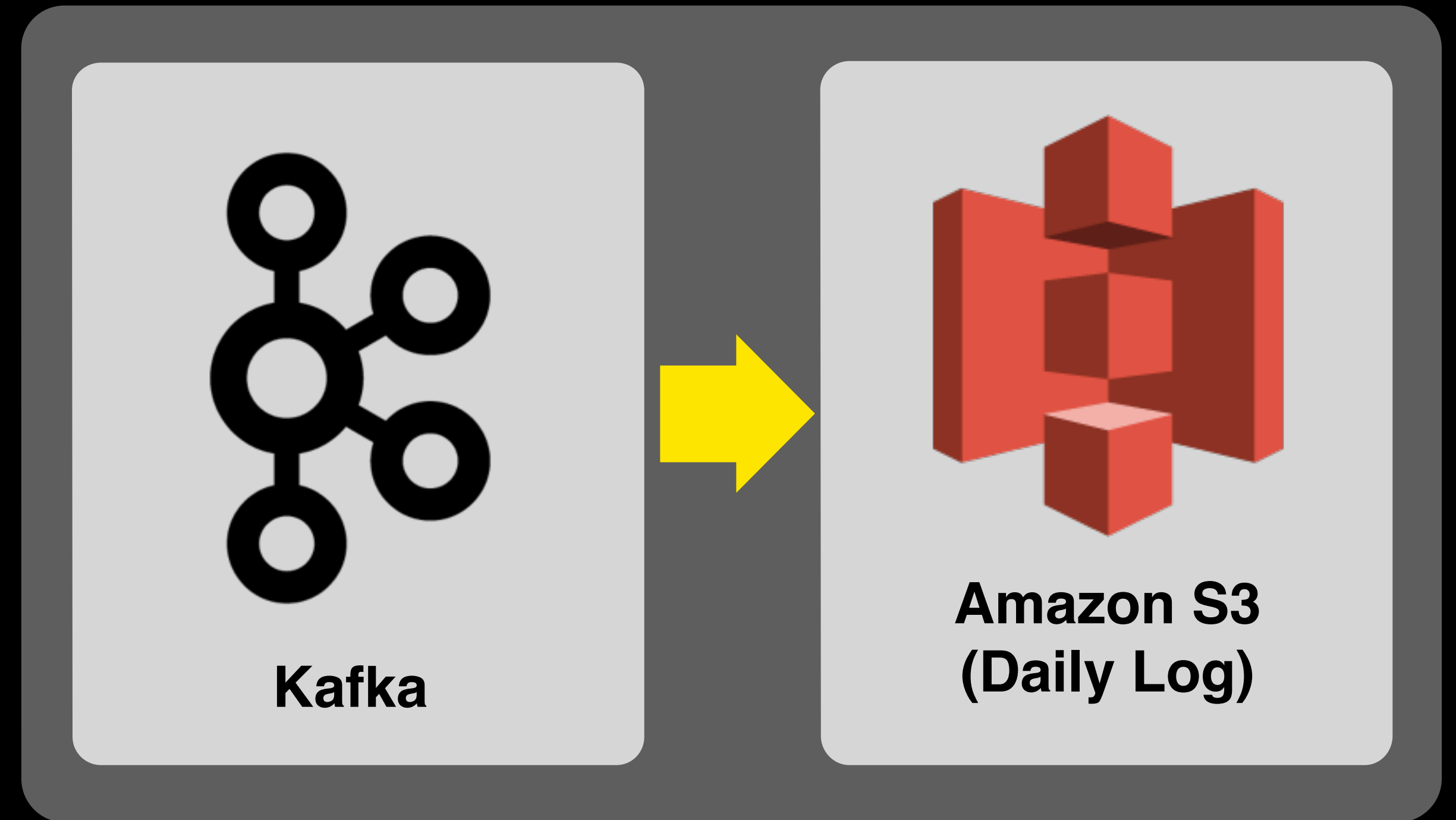
7. 로그 백업 Amazon S3

로그 백업

- 자기 테입 백업? → **바빠요**
- NAS? → **가난해요**
- MySQL + 일별 백업? → **바빠요**
- Hadoop 클러스터? → **가난해요**
- HBase / Cassandra 클러스터? → **바쁘고 가난해요**

7. 로그 백업 Amazon S3 아마존 S3

- 무한 용량
- 고 가용성
- 데이터를 밀어넣을 때의 용량당 비용 없음
- 저렴한 유지비
  - CPU 비용을 내지 않음

아마존 S3의 활용

- 파일 시스템이라 부르기엔 지원되는 기능이 몇 없다.
  - 파일이 한 번 만들어지면 수정이 안 된다.
- 파일의 부분 읽기를 지원함
  - 블럭 압축 및 인덱싱 가능
- 빠른 조회와 배치 분석이 가능한 포맷으로 보관

7. 로그 백업 Amazon S3

일별 백업

- Kafka로부터 어제치 로그 토픽을 덤프받는다.
- 어제 로그를 정렬한다.
  - 키: [Action, Member Id, Timestamp, Hash]
  - 원하는 데이터를 모여있게 한다.
  - 빠르게 중복제거할 수 있다.
  - 압축이 잘 된다.

블럭 압축

- Snappy Framed Format 사용
- 4MB 원본 블럭 단위로 압축
  - 압축 후 블럭 크기: 1MB 미만. 20%로 줄어든다.
  - 특정 일자의 내 플레이 기록을 열람하는 데에 1MB만 읽으면 된다.
- 압축 블럭을 이어붙이면 하나의 거대한 압축 파일
  - S3에서 다운로드받아 활용하기 편하다.

7. 로그 백업 Amazon S3

일별 백업

일별 백업 파일 구조

2015-05-19.sz

buy, id:10127, 10:12 am
buy, id:10127, 10:34 am
...

buy, id:91729, 10:12 am
login, id:10013, 01:13 am
...

...

play, id:10013, 10:14 am
play, id:10013, 10:23 am
...

2015-05-19.index.json

buy, id:10127, 10:12 am
buy, id:91729, 10:12 am
...
play, id:10013, 10:14 am

0.2초!

샘플링

- 로그 정렬 시에 Member id 를 역순으로 뒤집음
- 13100 → 00131, 193800 → 008391
- '00', '000' 으로 끝나는 member id의 로그가 모인다.
- 1%, 0.1% 샘플링된 로그를 빠르게 분석
  – 개발자 PC에서 빠른 데이터 어플리케이션 개발 및 테스트

로그 백업 Amazon S3 결과

- 특정 유저 하루치 플레이 로그 조회 0.2초
  - 어제 로그든 2년 전 로그든 같은 속도로
- 특정 유저 일년치 플레이 로그 조회 3초
  - Amazon EC2 c3.8xlarge, 32코어 기준

로그 백업 Amazon S3 결과

- 1/1000 샘플링 데이터 일주일치
  – 로컬에서 15초만에 분석
- 일년치 플레이 로그 전체
  – 20대 아마존 EMR 클러스터에서 30분 안에 분석
- 조회, 분석 모두 스케일 아웃 가능
  – 서버를 늘릴수록 성능이 빨라진다

로그 백업 Amazon S3

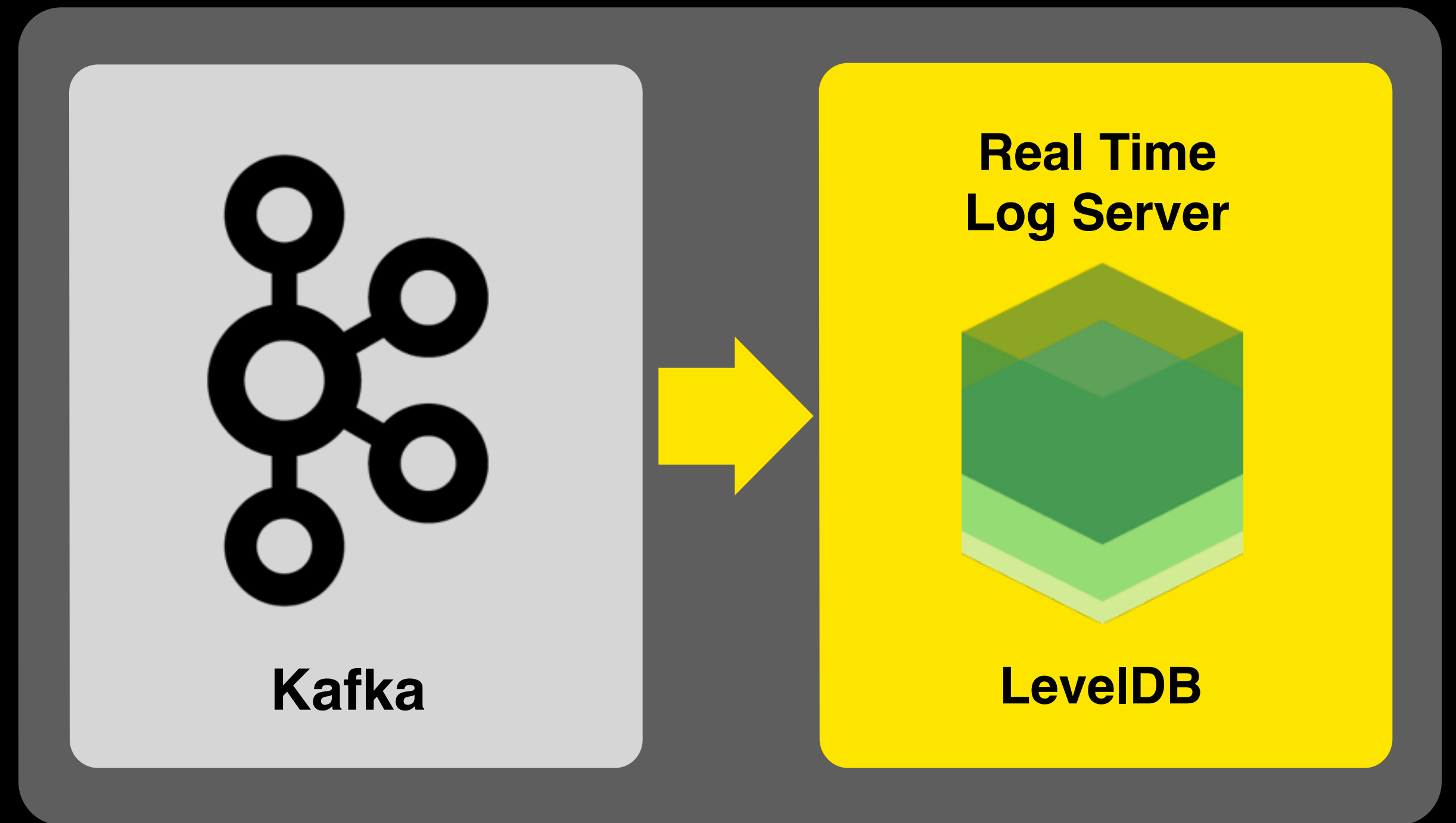
- 초기 투자비용 없음
- 2년치 전체 용량 20TB 내외
- 월 유지비 $700 미만
  - 사무실 전기세보다 싸다

실시간 로그 시스템

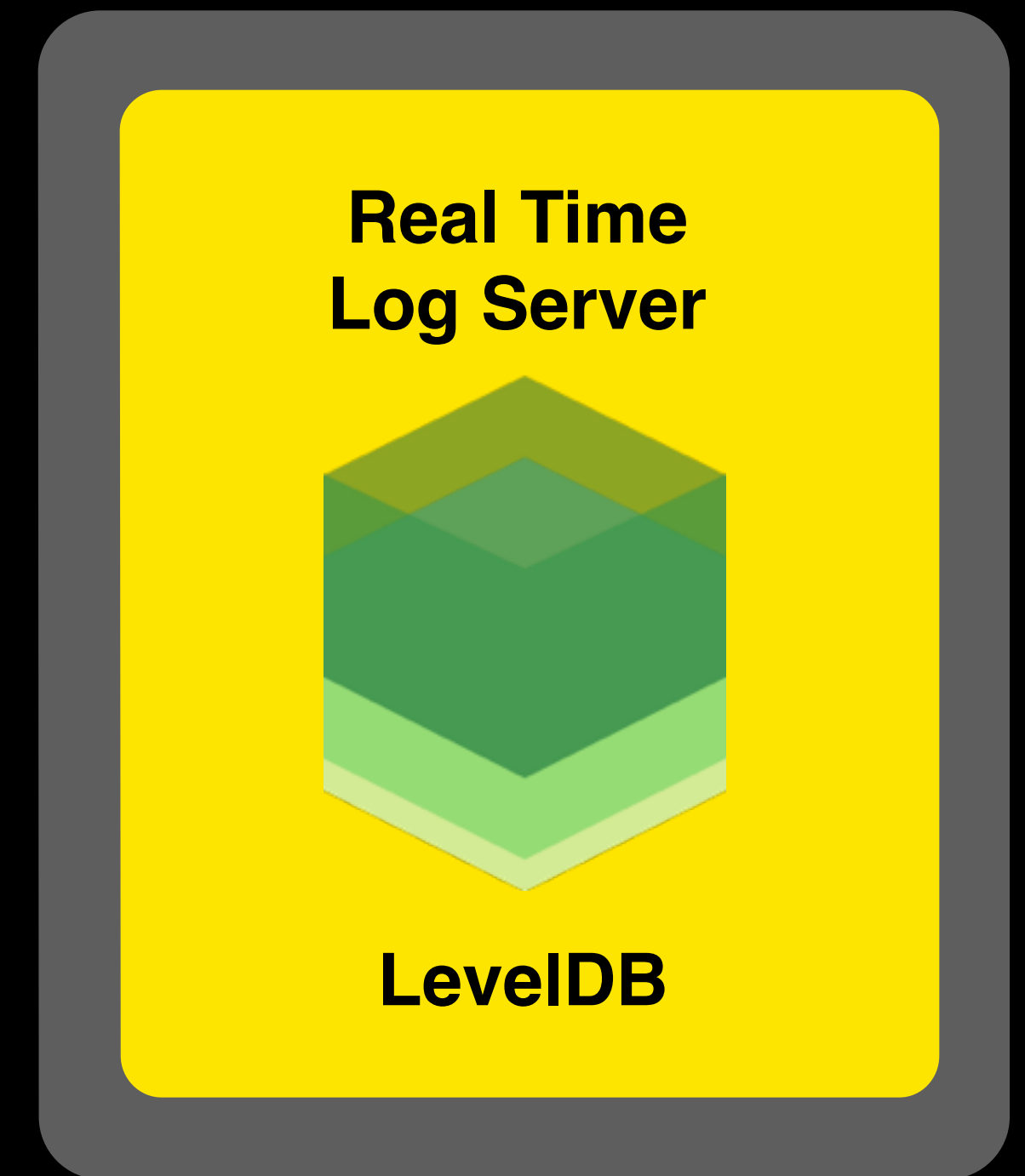
실시간 로그 조회

- CS를 위해서는 오늘 로그도 조회해야 한다.
- 어떤 시스템을 쓸 것인가?
  - 초당 읽기 / 쓰기 트랜잭션 수가 클 것
  - 범위 쿼리를 지원할 것
- 위 조건을 만족하는 시스템이 그다지 많지 않다.

실시간 로그 시스템 LevelDB

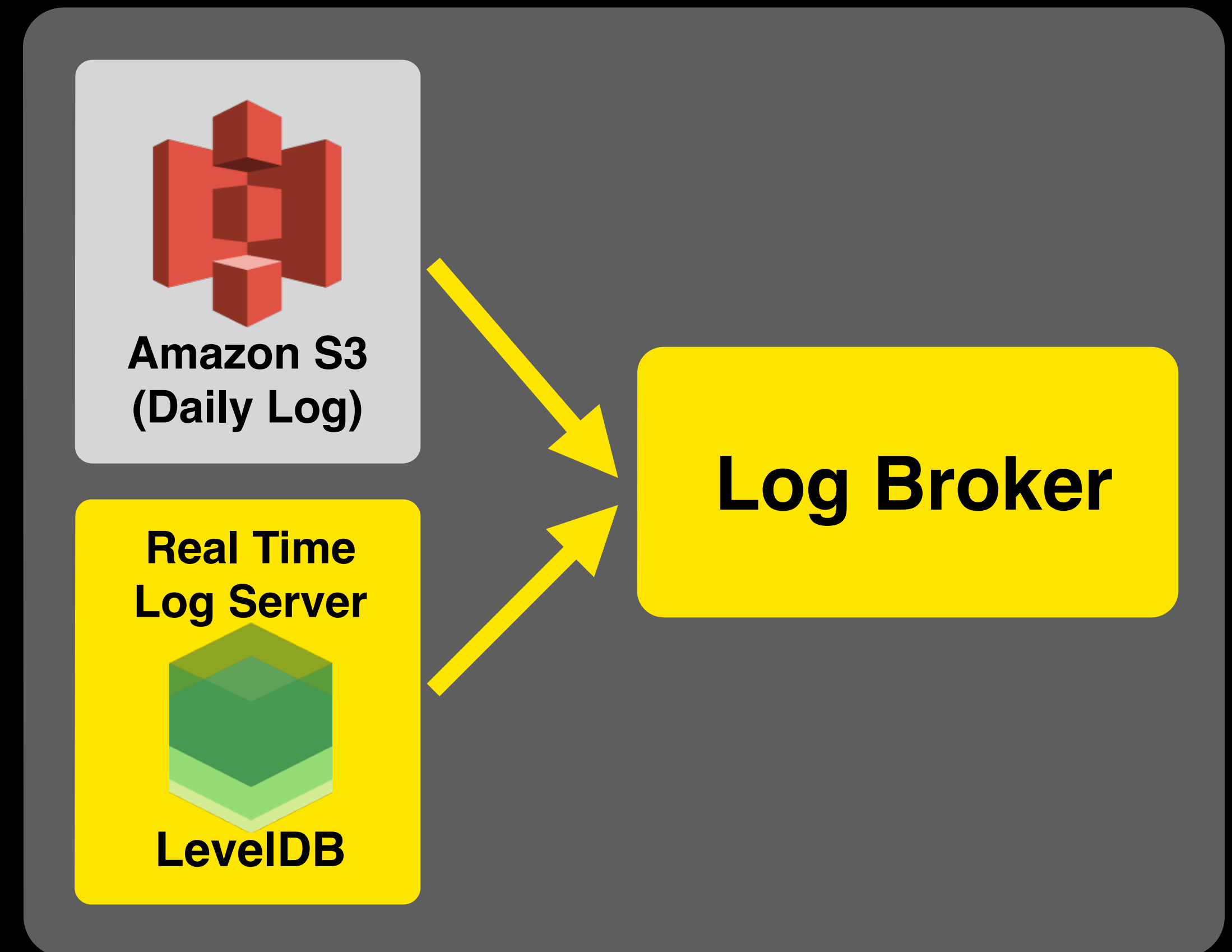
- Key-Value 스토리지
- 믿고 쓰는 구글 제품
  - 구글 크롬에 사용
- 범위 쿼리 지원
- 초당 10만건 쓰기 가능

실시간 로그 시스템 실시간 로그 서버

- LevelDB엔 네트워크 API가 없음
- REST API 서버를 Go로 작성
  - 높은 동시성
  - 멀티코어를 활용한 편한 스케일
- 일별 LevelDB
  - 일별 카프카 토픽에 대응
  - 중복 제거

로그 브로커

- Go로 작성
- 과거 로그는 S3
- 최근 로그는 실시간 로그 서버
- 동일한 인터페이스로 2년 전 로그와 1초 전 로그를 모두 접근 가능

로그 활용 예

로그 활용 예

사례: 플레이 롤백

- 특정 쿠키에 데이터 오류 발견
  - 비정상적으로 높은 점수 발생
  - 비정상 플레이로 인식되어 정지당하는 유저 발생
- 쿠키런은 기록 경쟁 게임, 공정함이 중요하다

로그 활용 예 플레이 롤백

- 오늘 플레이 검색으로 영향 받은 유저 확인
- 수백 명의 영향 받은 유저들에 대하여
  - 모든 플레이 기록을 가져와서
  - 문제 있는 플레이를 필터링하고
  - 문제 없는 플레이 중 최고 점수로 갱신
- 유저당 5초면 가능! → 세 시간만에 해결

로그 활용 예 CS 대응: 마카롱 웹 서비스

- 친절한 인터페이스로 로그를 보여준다
  - 크리스탈 입출
  - 아이템 입출
  - 플레이 기록
- 정확한 전후 상황을 판별하고 CS 대응 가능
  - 허위 CS 방지

로그 활용 예 데일리 레포트 : 팬케익

- Scalding 기반 로그 분석 프레임워크
  - 함수형 언어 문법을 사용한 간결한 코드
  - 샘플링을 활용한 빠른 로컬 테스트
- 매일 아침마다 어제 지표를 메일로 전송

로그 활용 예

Scalding을 활용한 분석

```
class ActiveUserCount(args : Args) extends Job(args) {
  CookierunLog("member/connect", args("sampling"),
    args("timeStart"), args("timeEnd"),
    Tuple1("memberSeq"))
  .groupBy('memberSeq){ _.take(1) }
  .groupAll{ _.size }
  .write(Csv(args("output")))
}
```

로그 활용 예 데일리 레포트 : 팬케익

- Amazon EMR 위에서 가동
  - 아마존이 요청에 따라 자동으로 띄워주는 Hadoop 클러스터
- Spot Instance 사용
  - 8 코어 서버 (c3.2xlarge) 15대 한 시간 빌리는 비용 $2 미만
  - 출근 교통비 수준

참고자료

- 홍성진, 쿠키런 1년 서버 개발 분투기, NDC 2014
- 이상곤, Python으로 쿠키런 운영하기, Pycon Korea 2014
- 김민우, 가볍고 유연하게 데이터 분석하기, KGC 2014
- 정창제, AWS 활용하여 Daily Report 만들기, AWS Global Summit Seoul 2015

함께 멋진 시스템을 만들어봐요!

We are hiring

career@devsisters.com

Q&A

감사합니다

DEVSISTERS